



**UNIVERSITY
OF OULU**

TIETO- JA SÄHKÖTEKNIIKAN TIEDEKUNTA

Eero Petteri Huttunen

Menetelmä violetin värivirheen poistoon valokuvista

Diplomityö
Tietotekniikan tutkinto-ohjelma
Toukokuu 2019

Huttunen Eero. (2019) Menetelmä violetin värivirheen poistoon valokuvista.
Oulun yliopisto, tietotekniikan tutkinto-ohjelma. Diplomityö, 40 s.

TIIVISTELMÄ

Violetti värivirhe on valokuvissa näkyvää virhettä, joka johtuu kameran ja linssin ominaisuuksista. Värivirheen määrää voidaan pienentää hankkimalla kalliimpia kameran komponentteja, mutta halvoissa laitteissa ei tämä ole vaihtoehto. Työssä tarkastellaan violetin värivirheen poistossa käytettyjä algoritmeja ja esitellään oma algoritmi. Oma algoritmi poistaa paremmin violettiä testikuvasta kuin yksi esitetyistä algoritmeista, mutta ei paremmin kuin kaupallinen Adobe Lightroom. Työssä tehtiin myös kysely, jossa osallistujia pyydettiin vertaamaan, mikä algoritmeista on parempi. Tulokset olivat vaihtelevia, joka kertoo värivirheen poiston haasteista.

Avainsanat: kamera, linssivirheet, kromaattinen aberraatio, digitaalinen kuvankäsittely.

Huttunen Eero. (2019) A method for removing purple fringing from photographs. University of Oulu, Degree Programme in Computer Science and Engineering. Master's Thesis, 40 p.

ABSTRACT

Purple fringing is purple tinted color error which is caused by cameras and its lenses errors. The amount of color error can be lowered by using a more expensive camera components, but in cheap devices this is not an option. In this work purple fringe removal algorithms are examined, and own implementation introduced. The implementation removes purple fringing from test photo better than one of the introduced algorithms but not better than commercial Adobe Lightroom. A survey was also made where participant where asked to choose which algorithm is better. The results were varying which tells about the challenges of purple fringe removal.

Key words: camera, lens errors, chromatic aberration, digital image processing.

SISÄLLYSLUETTELO

Tiivistelmä

Abstract

Sisällysluettelo

Alkulause

Lyhenteiden ja merkkien selitykset

1.	Johdanto.....	8
2.	Kamera	9
2.1	Kameran malli	9
2.2	Objektiivin eli kameraseläin.....	11
2.3	Digikamerasensorit.....	14
2.3.1	Mikrolinssi.....	16
2.4	Kameraseläin	16
3	Kameraseläin kuvantamisvirheet.....	18
3.1	Monokromaattinen aberraatio	18
3.1.1	Palloaberraatio.....	18
3.1.2	Koma	19
3.1.3	Hajataitto	19
3.1.4	Petzval-kentän kaarevuus	20
3.1.5	Vääristymät	20
3.2	Kromaattinen aberraatio	21
3.2.1	pitkittäinen aberraatio.....	21
3.2.2	Sivuttainen aberraatio.....	21
3.3	Pikselien ylivuoto	22
3.4	Interpolaatiovirheet.....	23
3.4	Violetti reunavirhe.....	23
3.5	Kuvanlaadun parantavia ja korjaavia algoritmeja	24
3.5.1	Kang	26
3.5.2	Soon-Wook Chung & al	26
3.5.3	Kim ja Park.....	28
3.5.4	Kannan ja Malik	29
3.5.5	Algoritmien yhteenveto	29
4	Toteutus	30
5	Tulokset.....	33

5.1 Tulosten ero mustavalkokuvasta	34
5.2 Tulosten arviointi kyselyllä	34
6 Keskustelu	36
7 YhteenVeto.....	37
8 Lähteet.....	38
9 Liitteet	40

ALKULAUSE

Haluaisin kiittää työn ohjaajaa Janne Heikkilää ohjeista ja kärsivällisyydestä työn teon aikana. Kiitän myös Markus Turtista, Matti Niskasta ja Jari Hannukselaa neuvoista ja opastuksesta.

Oulu, Toukokuu 27. 2019

Eero Petteri Huttunen

LYHENTEIDEN JA MERKKIEN SELITYKSET

RGB	Punainen, vihreä ja sininen väri, josta pikseli koostuu.
YCbCr	Väriavaruus, jossa Y on kuvan kirkkaus ja Cb ja Cr kuvaavat värin.
CMOS	Kuvasensorityyppi (eng. Conductive Metal-Oxide Semiconductor).
CCD	Kuvasensorityyppi (eng. Charge Couple Device)
f	Linssin polttoväli
R	Punainen värikanava. Pikselin punaisen värin määrä kuvassa.
G	Vihreä värikanava. Pikselin vihreän värin määrä kuvassa.
B	Sininen värikanava. Pikselin sinisen värin määrä kuvassa.

1. JOHDANTO

Värivirheet kameroissa ovat olleet aina ongelma. Jo ennen digitaalisen kameran keksimistä taittovirheitä on jouduttu korjaamaan kaukoputkista. Vanhat kaukoputket kärsivät sumeista kuvista, koska eri värit eivät tarkentuneet samaan paikkaan. Jopa ihmisen oman silmän virheitä joudutaan korjaamaan silmälaseilla. Linsseissä on aina jonkinlaista taittovirheitä, joiden suuruutta pyritään niiden suunnittelussa minimoimaan. Aina kaikkea virheitä ei saada korjattua, koska se maksaa liikaa.

Älypuhelinien määrä on viime vuosikymmenen aikana räjähtänyt ja melkein jokaisella on nykyään taskussaan semmoinen. Jokainen älypuhelin sisältää vähintään yhden kameran sisällään ja kamera ja sen kuvanlaatu ovat tulleet tärkeäksi osaksi, kun valitaan puhelinta. Kalliimmissa älypuhelimissa voi olla kaksi etukameraa kasvontunnistusta varten tai jopa kolme takakameraa, joilla jokaisella on erilainen linssi joko lähi- tai kaukokuvausta varten. Halvoissa puhelimissa ei tähän ole varaa, vaan täytyy jokaisesta osasta pihistellä. Säästäminen näkyy ensimmäisenä linsseissä, jotka eivät ole kehittyneet yhtä rajusti kuin kuvasensorit ja prosessorit, jotka ovat puhdasta elektroniikkaa. Halvat, muovista valmistetut linssit tuottavat värivirheitä, joiden takia tämä työ on tehty. Jos linssiä ei saada paremmaksi, täytyy kuvanlaatua parantaa toisella tavalla. Algoritmeilla parannetaan kuvanlaatua halvemmin kuin kalliilla osilla, joissa rahaa kuluu materiaaleihin, valmistukseen ja kuljetuksiin. Prosessorilla tehdään jo paljon kuvankäsittelyä ja linssin aiheuttamien virheiden korjaus on enää yksi osa prosessia.

Kahden puhelinvalmistajan eri puhelimilla voi olla käytännössä sama kamera ja linssi, mutta puhelimen sisäiset algoritmit ja jälkikäsittelyt tekevät kuvista täysin erilaatuisia. Esimerkiksi OnePlus 3T:llä ja Google Pixelillä on sama kamera, mutta Pixel tuottaa parempia kuvia esimerkiksi sen HDR+ algoritmin avulla. Tulevaisuudessa käytetyn elektroniikan vaikutus vähenee ja yhä enemmän kuvanlaadun määrää käytetty algoritmi.

2. KAMERA

Kamera on laite, jolla kaapataan valoa ja tallennetaan kuva maailmasta. Sitä on käytetty aikojen saatossa monien tapahtumien ja henkilöiden tallentamiseen ja on nykyaikana tärkeä osa elämää. Kameran toiminta perustuu täysin valon manipulointiin, ja ongelmat tulevatkin juuri valon ominaisuuksista. Kameran linssi joutuu ottamaan huomioon eri värien eri aallonpituudet, valotusaukko valon diffraktion ja sensorit, jotka ovat mikroskooppisen pieniä, eivät enää kerää tarpeeksi valoa. Kameran vaatimukset muuttuvat isoista avaruuskaukoputkista pieniin älypuhelimien kameroihin ja aina ei voida saavuttaa ideaalista kuvaa. Esimerkiksi pienissä kameroissa ei voi olla kovin pitkää linssiä tai isoa sensoria, jolloin syntyy kohinaa ja taittovirheitä. Tässä kohdassa keskitytäänkin näihin pieniin kameroihin, jotka keräävät näkyvää valoa (380 – 700 nm).

2.1 Kameran malli

Kamera tallentaa kolmiulotteisen maailman kaksiulotteiseen tasoon eli tehdään niin sanottu perspektiiviprojektio. Kuvasta 2 näkee, kuinka piste maailmassa projisoidaan piirtämällä viiva koordinaatiston origoon, jolloin saadaan uusi piste viivan ja tason leikkauspisteessä. Leikkauspisteet p' saadaan kertomalla piste p maailmassa kameramatriisilla K ,

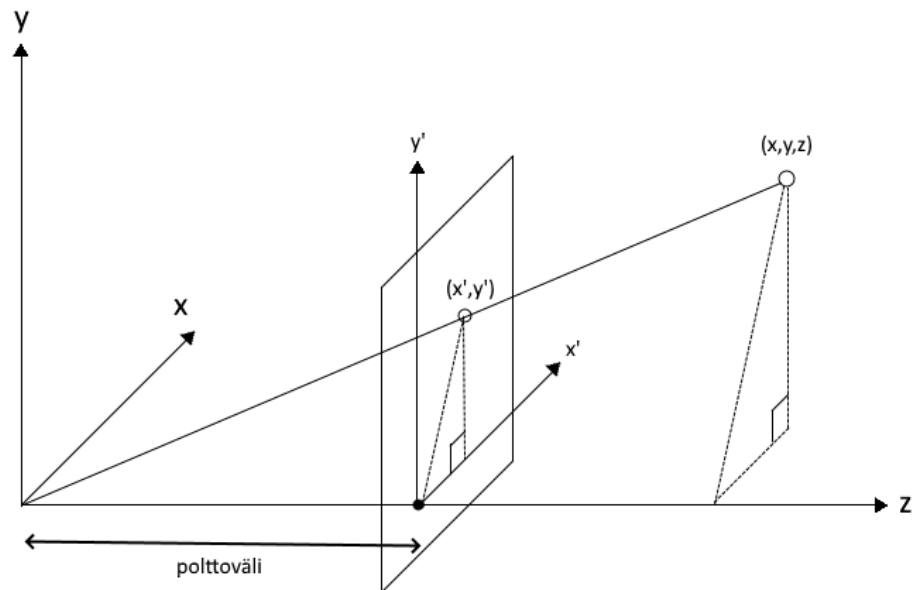
$$K = \begin{bmatrix} f & s & c_x \\ 0 & af & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$p = [x, y, z]$$

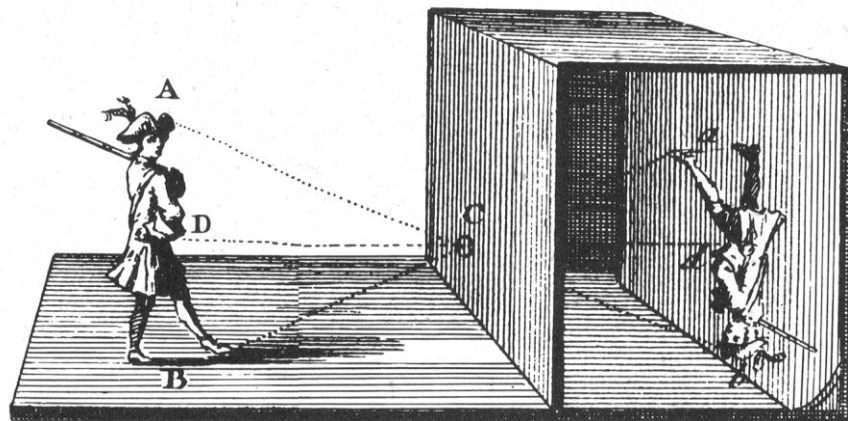
$$p' \sim p * K$$

jossa f on kameran polttoväli ja (c_x, c_y) on optinen keskipiste eli piste, jossa taso ja optinen akseli (tässä tapauksessa z -akseli) leikkaavat toisensa. Muuttuja s on kuvan vinous, joka tulee, kun taso, johon projisoidaan, ei ole kohtisuorassa optisen akselin kanssa. Koska taso ei välttämättä ole neliö, polttoväli f voi olla eri horisontaaliselle ja vertikaaliselle suunnalle. Jos toinen polttoväli tiedetään, voidaan toinen laskea kertomalla muuttujalla a , joka on kuvan kuvasuhde eli tason leveyden ja korkeuden välinen suhde. Saadut pisteet ovat suoraan verrannollisia (merkataan merkillä \sim) kameramatriisin ja pisteen kertolaskuun.

Neulanreikäkamera eli camera obscura (kuva 2) on esimerkki perspektiiviprojektioista: valo kulkee pienen aukon kautta ja projisoi kuvan maailmasta väärinpäin origon, eli neulanreikäkamerassa aukon, toiselle puolelle. Origin ja esineen väliin projisoituvaa kuvaa, kuten kuvan 1 tasoa, kutsutaan valekuvaksi tai virtuaalikuvaksi, koska sitä ei voida projisoida pinnalle kuten väärinpäin projisoitua kuvaa.



Kuva 1. Perspektiiviprojektio. Piste (x, y, z) projisoidaan kuvatasoon, jolloin syntyy piste (x', y') . Taso on polttovälin päässä origosta.

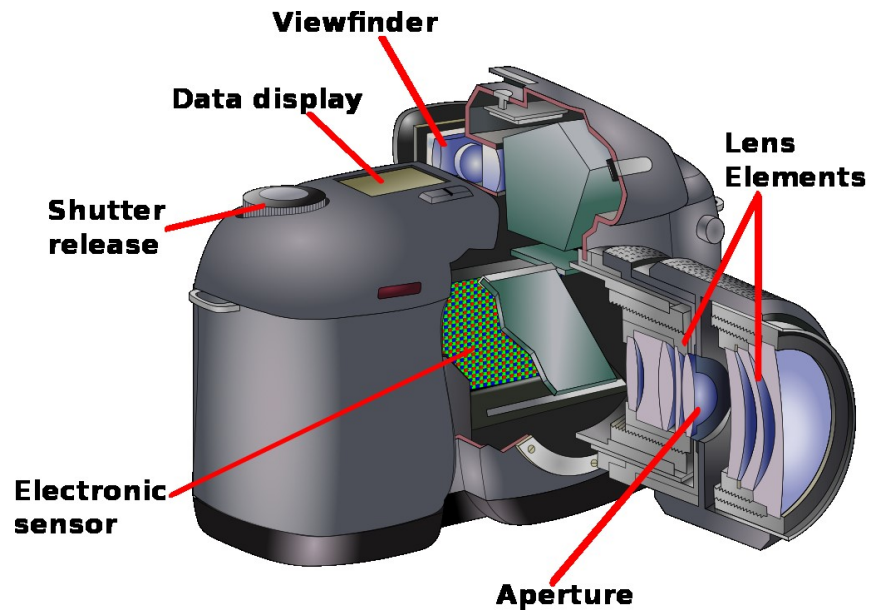


Kuva 2. Camera Obscura. ¹

Jos neulanreikäkameraa verrataan kuvan 3 oikeaan kameraan, periaate on sama, mutta aukon eteen laitetaan linssi ja seinän tilalla on sensori tai filmi, johon kuva kaapataan. Tulevan valon määrää ja tarkennusta kohennetaan linssin avulla. Kameran aukko määrää, kuinka paljon valoa kerätään kuvanoton aikana. Pieni aukko kerää vähemmän valoa, jolloin kuvan ottamiseen tarvitaan enemmän aikaa. Pitkillä valotusajoilla nopeat esineet sumentuvat, mutta kohinan määrä pienenee. Järjestelmäkameroissa on mekaaninen aukko, jolloin aukon kokoa voidaan säädellä, mutta esimerkiksi useimmissa älypuhelimissa puuttuu tämä ominaisuus, jolloin aukon koko on vakio.

¹ Lähde: File:001 a01 camera obscura abrazolas.jpg. (2016, October 27). Wikimedia Commons, the free media repository. Retrieved 18:11, May 27, 2019 from https://commons.wikimedia.org/w/index.php?title=File:001_a01_camera_obscura_abrazolas.jpg&oldid=211004883.

Lopuksi kuva tallennetaan joko filmille (käytetään enää lähinnä harrastuksissa) tai useimmiten digitaaliselle sensorille.



Kuva 3. Digitaalinen kamera².

2.2 Objektiivi eli kameran linssi

Kameran objektiivi on linssijärjestelmä, jolla valo saadaan tarkennettua pintaan. Neulanreikäkameran rajoituksena on reiän koko [1, 2]. Liian pieni reikä ei päästä paljon valoa läpi ja valon diffraktio tulee ongelmaksi. Liian suuri reikä taas tekee kuvasta epätarkan. Linssillä saadaan kerättyä enemmän valoa ja pidetään kuva tarkkana. Linssien käytöstä tulee myös omat ongelmansa.

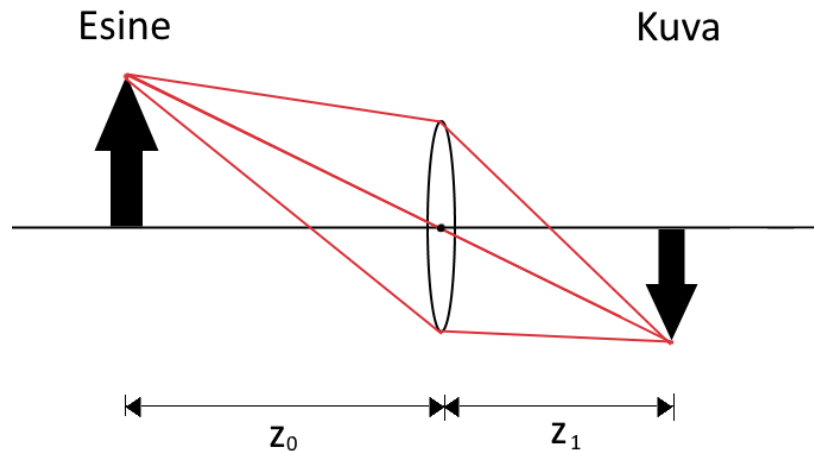
Yksinkertaisimmillaan linssijärjestelmä on kapean linssimallin mukainen [3]. Linssin voidaan tässä ajatella olevan ideaalinen neulanreikä, jonka kautta valo taittuu kohti polttopistettä. Kuvasta 4 näkee, miten valonsäteet esineestä tarkentuvat kuvaksi linssin toisella puolella. Kuva muodostuu polttovälin päähän linssistä, jolloin polttoväli voidaan helposti laskea esineen ja kuvan etäisyyksistä linssistä käyttämällä kaavaa:

$$\frac{1}{z_0} + \frac{1}{z_1} = \frac{1}{f}$$

Liikuttamalla linssiä voidaan z_1 :tä pienentää tai suurentaa, jolloin saadaan tarkennettua kohteisiin, jotka ovat kauempana tai lähempänä kameraa. Polttovälin muuttaminen muuttaa myös kuvan näkökenttää ja suurennosta [1]. Pienellä polttovälillä näkökenttä on laajempi ja suurennos pieni, joten se soveltuu kalansilmälinssisiin, joiden näkökenttä on 65° tai suurempi. Suuri polttoväli

² Lähde: Wikipedia contributors. (2019, April 14). Camera. In Wikipedia, The Free Encyclopedia. Retrieved 18:29, April 14, 2019, from <https://en.wikipedia.org/w/index.php?title=Camera&oldid=892450147>

päinvastoin tuottaa pienen näkökentän ja suurentaa kuvaa, joka sopii hyvin teleobjektiiveihin, joiden näkökenttä on 35° tai pienempi.



Kuva 4. Kapea linssimalli. Z_0 on etäisyys esineeseen linssistä ja Z_1 etäisyys linssistä kuvaan.

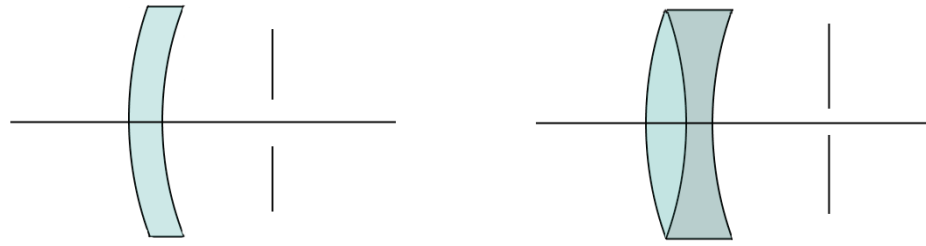
Linssit voidaan jakaa kahteen kategoriaan: refraktiolinssit ja peililinssit. Refraktiolinssit [1, 4] ovat lasisia, kaarevia linsejä, jotka päästävät valon lävitseen ja samalla taittavat valon niin, että se tarkentuu kuvatasoon. Peililinssit [4] sen sijaan heijastavat valon tasoon sitä taittamatta. Peililinssejä ei nähdä paljon valokuvauksessa, koska himmentimen aukon kokoa eli tulevan valon määrää ei voi säätää muulla kuin valotusajalla.

Linssejä on keksitty aikojen saatossa monenlaisia ja niillä on kuvanoton yhteydessä tarkoitus minimoida taittovirheet. Linssien suunnittelijat pyrkivät optimoimaan linssit monenlaisilla muuttujilla, kuten linssien etäisyyksillä toisistaan, linssien kaarevuuksilla, eri materiaaleilla tai linssien määrällä. Linssit kootaan peruspalikoista ja optimoidaan tarkoitusta varten. Käydään siis läpi muutamia linssityyppejä.

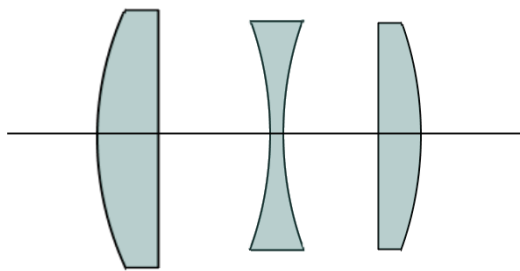
Yksittäinen kaareva linssi on yksinkertaisin linssisysteemi, jota käytetään halvoissa kameroissa tai valosensorissa. Linssi tarvitsee erillisen aukon (kuten camera obscurassa), joko linssin edessä tai takana. Tulos on yleensä huonolaatuista ja kärsii pallo- ja väriaberraatiosta. Virhettä voidaan vähentää tekemällä linssin kummastakin puolesta kaareva, jolloin saadaan meniscus-linssi.

Kromaattista aberraatiota voidaan vähentää liittämällä linsejä (toinen kovera ja toinen kupera) toisiinsa, jolloin saadaan akromaattinen linssi (Kuva 5). Linssit tehdään eri materiaalista, jotta niiden taittokerroin on eri. Näin saadaan punainen ja sininen aallonpituus kohdistettua samaan paikkaan.

Kuvan 6 Cooke-kolmoislinssi koostuu kolmesta linssistä: yksi kovera linssi kahden kuperan linssin välissä. Se on vanha linssi, patentoitu vuonna 1893, ja on nykyään ehkäpä käytetyin linssi joko sellaisenaan tai muokattuna versiona. Cooke-kolmoislinssi mahdollistaa kaikkien yleisempien aberraatioiden poistamisen. Linssisysteemissä on kahdeksan muuttujaa, joilla tulosta voidaan optimoida: kuusi linssin kaarevuutta ja kaksi linssin etäisyyttä toisistaan. Linssien paksuus ei vaikuta paljoa aberraatioiden poistossa. Huomioitavaa on, että kaikkea aberraatiota ei saada poistettua ja linssit pitää optimoida tietyille polttovälille ja näkökentälle.



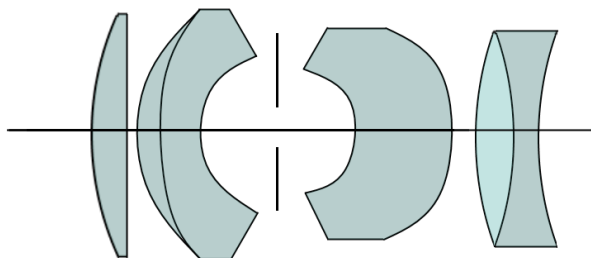
Kuva 5. Yksinkertainen meniscus-linssi ja aukko. Oikealla akromaattinen linssi.



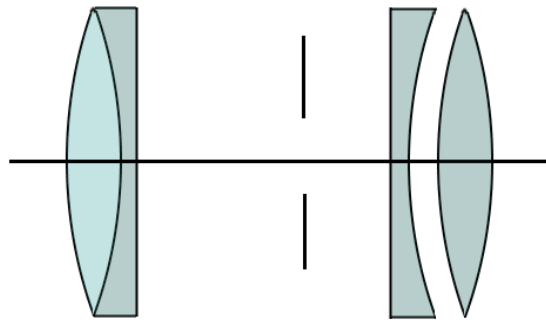
Kuva 6. Cooke-kolmoislinssi.

Gauss-kaksoislinssi saa nimensä kahdesta Gauss-linssistä. Kuvasta 7 nähdään, että Gauss-linssi on kaksi meniscus-linssiä, positiivinen linssi objektiivin puolella ja negatiivinen kuvapuolella. Gauss-kaksoislinssissä laitetaan yhden negatiivisen ja positiivisen linssin sijasta kaksi positiivista ja negatiivista linssiä peräkkäin, jolloin valon taitto jaetaan monelle linssille ja vähennetään näin vääristymiä. Saatu linssi on tehokas linssi, jonka ainoana heikkoutena on pallovääristymä. Gauss-kaksoislinssiä on käytetty useimpien 35mm objektiivien perustana ja siitä voidaan muokata laajakuvalinssejä sekä sopimaan pienille valotusajoille.

Petzval-linssiä kehitettiin lyhyitä valotusaikoja varten, kuten muotokuvien ottoon. Nykyään sitä käytetään liikkuvien kuvien otossa. Kuvan 8 Petzval-linssi koostuu yksinkertaisimmillaan kahdesta akromaattisesta linssistä, joista etummaisella linssillä on kaksinkertainen polttoväli systeemiin verrattuna ja takimmaisella linssillä sama polttoväli kuin systeemillä. Tämä systeemi mahdollistaa vähennetyn kromaattisen aberration, mutta vain pienelle näkökentälle eikä siis sovellu laajakuvien ottamiseen.

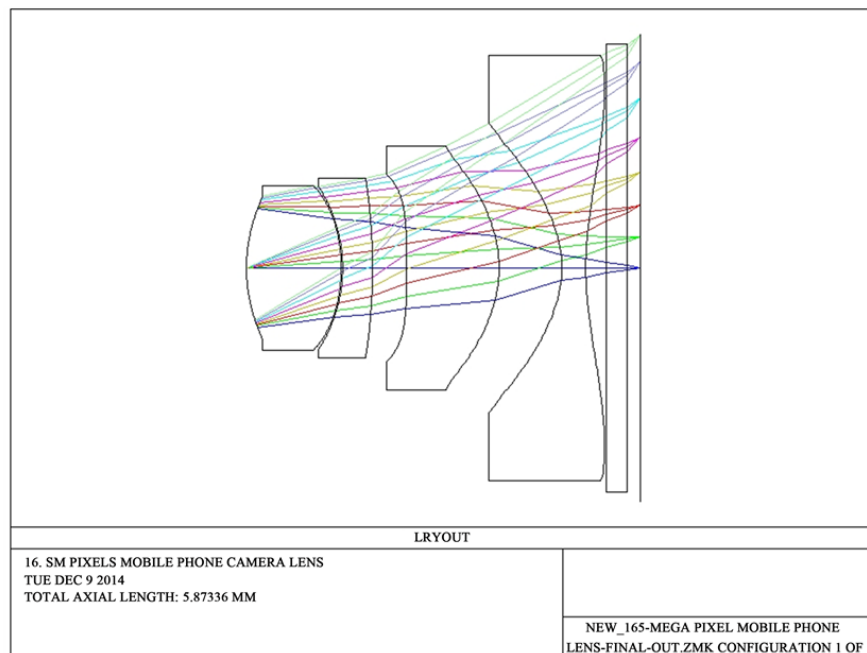


Kuva 7. Gauss-kaksoislinssi (biotar).



Kuva 8. Petzval-linssi.

Otetaan esimerkiksi vielä kuvan 9 valmis linssimalli [5]. Tämä linssi on suunniteltu mobiililaitteille ja koostuu kolmesta muovisesta asfäärisestä linssistä, yhdestä lasisesta pallomaisesta linssistä ja infrapunasuodattimesta ennen sensoria.



Kuva 9. Linssi mobiililaitteille³.

2.3 Digikameran sensorit

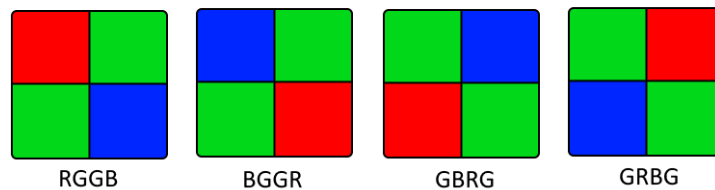
Kun kuva on saatu tarkennettua, se pitää jotenkin tallentaa. Kemialliset menetelmät kuten filmi olivat pitkään ainoa keino tallentaa kuvia, kunnes digitaaliset kamerat keksittiin ja yleistyivät. Digitaalisen sensorin etuna on, että kuvaa ei tarvitse kehittää

³ Lähde: Ma, Yuke, and V. Borovytsky. "Design of a 16.5 megapixel camera lens for a mobile phone. *Open Access library journal* 2.3 (2015). Retrieved 20:03, 12 May 2019 from http://file.scirp.org/Html/68103_68103.htm Available via license: CC BY 4.0

ja tuloksen näkee heti, joten filmikameroita ei enää nähdä muilla kuin harrastelijoilla tai osalla ammattikuvaajilla.

Filmin tilalle keksittiin ensin CCD (Charge Couple Device) sensori ja sitten CMOS (Conductive Metal-Oxide Semiconductor) sensori [6]. Vaikka CCD ja CMOS sensorit keksittiin samoihin aikoihin 1960-luvun loppupuolella, CCD -sensoreiden valmistus oli siihen aikaan helpompaa ja ne tarjosivat paremman kuvanlaadun ja vähemmän kohinaa. Vasta 1990-luvulla pystyttiin valmistamaan tarpeeksi hyviä CMOS -sensoreita ja nykyään ne kilpailevat CCD -sensoreiden kanssa kuvanlaadun hyvyydessä. CCD -sensoreissa haittapuolena on niiden valon vuotaminen muihin alueisiin. Hyvin kirkas valo voi vuotaa toisiin pikseleihin, jolloin kuvaan syntyy joko pystysuuntaisia viivoja (eng. smearing) tai valo vuotaa kaikkiin suuntiin (eng. blooming). Nykyisissä CCD -sensoreissa otetaan nämä mahdolliset virheet huomioon ja sensorit suunnitellaan niin, että elektronien ylivuotoa ei tapahdu niin paljon. CMOS -sensorit ovat luontaisesti hyvin kestäviä tälle ylivuodolle, jolloin virheet jäävät yleensä hyvin vähäiseksi.

Kummankin sensorin toiminta perustuu valon vuorovaikutukseen sensorin fotodiodin kanssa. Fotodiodit reagoivat kaikkiin näkyvän valon aallonpituuksiin eli toisin sanoen se on monokromaattinen ja sellaisenaan käytettynä sillä saadaan otettua vain mustavalkoisia kuvia. Jotta kuvasta voitaisiin luoda RGB värikuva, tarvitaan pikselille punainen, vihreä ja sininen väri. Värit voidaan erotella esimerkiksi prismalla, joka taittaa jokaisen värin omalle sensorilleen, jolloin tarvitaan kolme sensoria tallentamaan kuvan. Halvempi tapa on laittaa sensorin päälle värifiltteri, joka suodattaa kaiken paitsi halutun värin aallonpituudet. Tätä kutsutaan värifiltteritaulukoksi (eng. Color Filter Array), jossa jokainen fotodiodi kerää tiettyä aallonpituusalueutta. Sekä CMOS että CCD -sensorytyypissä pikselit on yleensä aseteltu Bayer-mallin [6, 7, 8, 9] mukaisesti eli sensorin 2x2 alueella on yksi R (punainen), kaksi G:tä (vihreä) ja yksi B (sininen) sensori. Nämä värit voivat olla aseteltuna eri tavoilla riippuen valmistajasta. Kuvasta 10 näkee eri Bayer-mallin mahdolliset asetelmat.

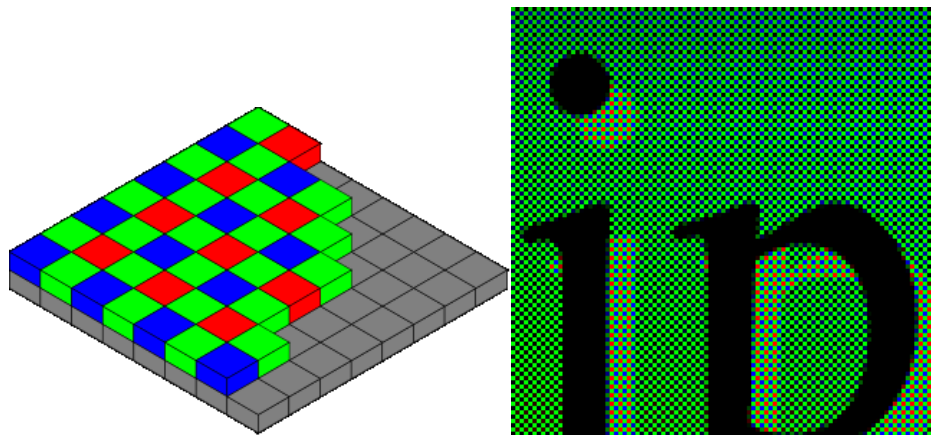


Kuva 10. Bayer mallin erilaiset asetelmat.

Sensorytyyppejä on toki erilaisia [6, 9]. R, G ja B voidaan esimerkiksi asetella pseudosatunnaisesti, jolloin värit eivät ole aseteltu symmetrisesti. Värit voidaan myös asetella niin, että jokainen väri on omalla rivillään tai sarakkeellaan, joka sopii hyvin televisiota varten. RGB tyyppin tilalla voi olla CMY tyyppi, jossa R, G ja B:n tilalla on Cy (syaani), Mg (magenta) ja Y (keltainen). Tämän "YMgCy" formaatin voi johtaa RGB:stä vähentämällä valkoisesta valosta (W) jokin väreistä R, G tai B.

$$\begin{aligned} Y &= R + G = W - B \\ Mg &= R + B = W - G \\ Cy &= G + B = W - R \end{aligned}$$

Suurin osa sensoreista on kuitenkin Bayer-mallista, joten keskitytään siihen. Sensorilta saatua kuvaa kutsutaan raakakuvaksi. Raakakuvaa ei voida näyttää tavallisessa näytössä, koska jokaisella pikselillä on vain yksi väri ja RGB kuva vaatii kolme väriä per pikseli. Kuvassa 11 raakakuvaa on visualisoitu asettamalla puuttuvat värit nolllaksi. Jotta saadaan kokonainen RGB-kuva, täytyy puuttuvat värit interpoloida ympäröivistä pikseleistä. Interpolointimenetelmiä on paljon ja eri valmistajat käyttävät eri menetelmiä vaatimusten mukaan.



Kuva 11. Bayer-malli. Vasemmalla sensori ja oikealla raakakuvan visualisointi.⁴

Yksi helpoimmista menetelmistä on tavallinen bilineaarinen interpolaatio, joka aiheuttaa virheitä reunoissa, joissa väri muuttuu terävästi. Tällöin kahden eri värin väliin interpoloidaan näiden kahden värin keskiarvo, joka ei esiinny alkuperäisessä kuvassa. Siksi laadukkaammissa interpoloinneissa [7, 9] otetaan huomioon kuvan reunat ja gradientit, joiden avulla voidaan paremmin laskea kuvan oikeat värit.

2.3.1 Mikrolinssi

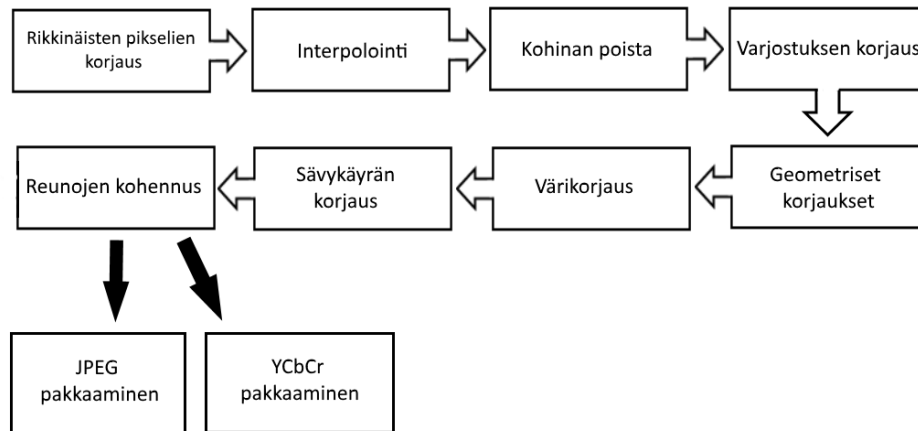
Jotta kuvasta saadaan tarkka, pitää kuvan resoluutio olla suuri. Yksi keino olisi kasvattaa sensorin kokoa, mutta se ei ole aina vaihtoehto, kun kamerasensorin pitäisi olla pieni. Resoluutiota kasvattaakseen pitää siis pikselien kokoa pienentää, jotta samalle sensorille mahtuu enemmän pikseleitä. Tämä johtaa siihen, että pikseleille ei riitä valoa, jolloin tarvitaan mikrolinssi pikselin lähellä, joka kerää valon sensorille. Mikrolinssit pienentävät samalla myös ylivuodon mahdollisuutta, koska valo kohdennetaan oikeaan paikkaan [6].

2.4 Kameraputki

Sen jälkeen, kun kamerasensorilta on saatu raakakuva, sitä voidaan jälkikäsitellä vähentämällä kohinaa, korjaamalla värejä tai pakkaamalla se toiseen tiedostomuotoon

⁴ Lähde: Wikipedia contributors. (2019, February 9). Demosaicing. In *Wikipedia, The Free Encyclopedia*. Retrieved 18:34, April 14, 2019, <https://en.wikipedia.org/w/index.php?title=Demosaicing&oldid=882455775>

[10, 11]. Kaikissa tavallisissa digitaalisissa kameroissa käydään tämä putki läpi kutakuinkin kuvan 12 tapaan.



Kuva 12. Android-älypuhelimien kameran kuvaputki ja sen vaiheet.

Ensimmäiseksi korjataan pikselit, jotka ovat rikki ja antavat aina lukemaksi joko nollan (kuollut pikseli) tai suurimman mahdollisen arvon (kuuma pikseli). Pikselin arvo voidaan korjata ottamalla ympärillä olevista keskiarvo tai yksinkertaisesti kopioimalla lähin pikseli. Interpolointi käytiin jo kohdassa 2.3 läpi. Puuttuvat RGB arvot interpoloidaan ympäristöstä. Kohinan poistoon löytyy monenlaisia menetelmiä, mutta monesti ongelmana on reunojen liiallinen tasoittaminen ja yksityiskohtien sumentuminen. Linssi voi aiheuttaa kuvassa vinjettiä, jolloin kuvan nurkat ja sivut ovat kuvan keskikohtaa tummempia, tai geometrisia muutoksia, kuten tynnyrivääristymää (katso kohta 3.1.5), joten ne korjataan tässä vaiheessa. Riippuen kuvattavasta kohteesta, kuvalle tehdään väri- ja sävykäyrän korjaukset. Kuvan värit korjataan valkotasapainolla, jotta valkoiset kohteet näkyisivät valkoisina lopullisessa kuvassa ja sävykäyrää muuttamalla voidaan säätää värisaturaatiota. Ennen kuin kuva laitetaan eteenpäin, terävöitetään yksityiskohtia ja reunoja, jotka kohinan poiston aikana saattoivat sumentua. Tosin tämä kasvattaa taas kohinan määrää.

Lopuksi kuva pakataan pienempään tilaan joko koodaamalla se JPEG muotoon tai YCbCr väriavaruuteen, joka kehitettiin alun perin videota varten. YCbCr-kuvassa kuva jaetaan luminanssiin (kuvan kirkkaus) ja kromaaniin (väri). Luminanssi tallennetaan Y kuvaan, joka vastaa harmaasävykuvaa. Kromassa on kaksi arvoa, Cb ja Cr, jotka ovat koordinaatteja välillä $[-0.5, 0.5]$ ja osoittavat värin CbCr-kartassa. Esimerkiksi jos Cb ja Cr ovat kummatkin -0.5 , on kuvan pikseli siinä kohdassa vihreä. Tilaa säästääkseen voidaan 2×2 kokoiselle alueelle käyttää samaa Cb:ta ja Cr:ta eli kromaa näytteistetään. Näytteiden määrää voidaan vaihdella ja tämä on vain yksi vaihtoehto. Lopputuloksena CbCr osa on puolet pienempi kuin Y ja koko YCbCr on puolet pienempi kuin RGB kuva.

3 KAMEROIDEN KUVANTAMISVIRHEET

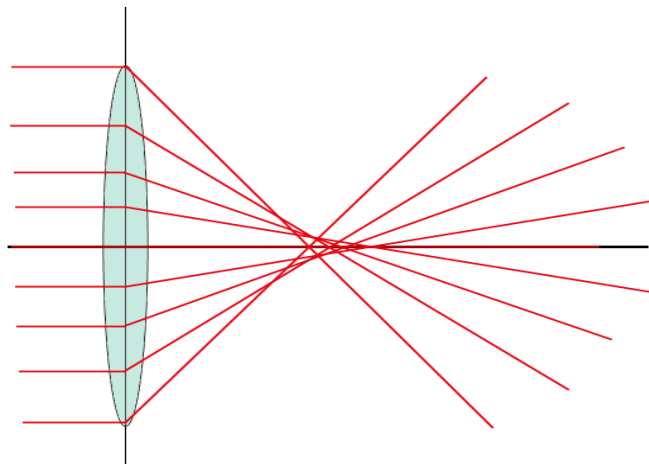
Koska linssien valmistus ei ole ikinä täydellistä, linssisysteemit tuottavat aina jonkinlaista virhettä kuvaan. Linssit suunnitellaan käyttäen ensimmäisen asteen teoriaa eli valonsäteet ajatellaan täydellisiksi viivoiksi eikä valon aalto-ominaisuutta oteta huomioon. Tämä approksimointi tuottaa yleensä tarpeeksi hyvän tuloksen, mutta aiheuttaa virheitä eli aberraatiota [12, 13, 14, 15, 16], jotka linssien suunnittelussa täytyy ottaa huomioon. Aberraatiot voidaan jakaa kahteen osa-alueeseen: kromaattiseen ja monokromaattiseen aberraatioon. Kromaattiset aberraatiot ovat värivirheitä, joissa eri aallonpituudet taittuvat linssissä eri tavalla, jolloin värit eivät kohdistu samaan pisteeseen. Monokromaattiset aberraatiot tapahtuvat aallonpituudesta huolimatta.

3.1 Monokromaattinen aberraatio

Monokromaattinen aberraatio saa nimensä siitä, että sitä tapahtuu vaikka valonsäteet olisivat samaa aallonpituutta eli yksiväristä. Monokromaattista aberraatiota on palloaberraatio, koma, hajataitto, Petzval-kentän kaarevuus ja vääristymät.

3.1.1 Palloaberraatio

Palloaberraatiossa hyvin kaukaa tulevat valonsäteet, joiden voidaan olettaa olevan yhdensuuntaisia linssin optisen akselin kanssa, eivät kohdistu samaan horisontaaliseen polttopisteeseen osuessaan linssin eri osiin (kuva 13). Aberraatio voi olla ”positiivinen”, jolloin linssin optisesta akselista kauimmaiset säteet kohdistuvat lähemmäs linssiä kuin optisen akselin lähellä olevat säteet. Negatiivisessa palloaberraatiossa säteet kohdistuvat päinvastoin kauemmas linssistä.

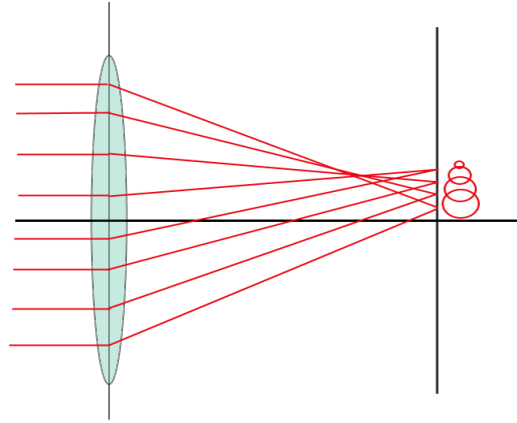


Kuva 13. Positiivinen palloaberraatio.

Palloaberraatio aiheuttaa sumeutta ja kontrastin menetystä kuvissa. Kuvan pisteiden ympärille syntyy rengasmainen artefakti, jolloin näyttää kuin valopisteet vuotaisivat valoa ympäristöönsä.

3.1.2 Koma

Jos oletetaan taas valonsäteiden tulevan hyvin kaukaisesta objektista, komassa reunimmaisiet valonsäteet tarkentuvat eri paikkaan kuin linssin keskeltä kulkevat säteet. Toisin kuin palloaberraatioissa, säteet kohdistuvat eri paikkaan vertikaalisessa suunnassa.

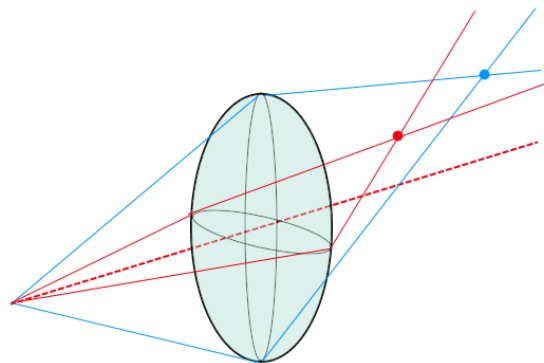


Kuva 14. Koma-aberraatio.

Koma-aberraatio tuottaa kuvassa komeettamaista virhettä (josta nimi tulee), joka näkyy pyrstömäisenä virheenä pisteissä. Koman voi sanoa olevan suurennoksen muutos, koska se luo pienempiä tai suurempia kuvia kohteesta. Kuvasta 14 näkee, kuinka isoimman ympyrän lisäksi syntyy pienempiä ympyröitä isoimman ympyrän viereen (komeettamuoto).

3.1.3 Hajataitto

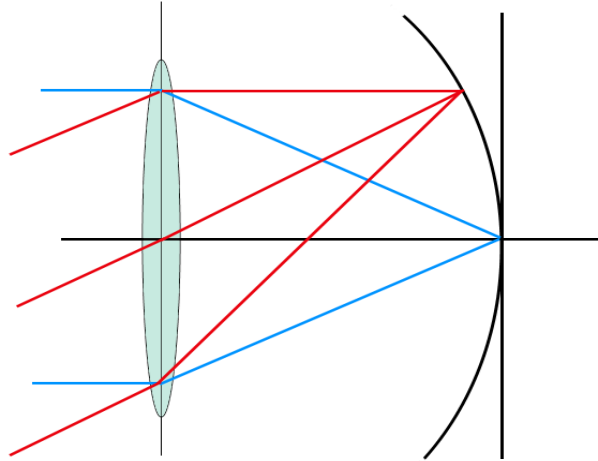
Hajataitossa tai astigmatiassa horisontaalinen ja vertikaalinen polttopiste eivät ole samassa kohdassa ja polttopisteet päätyvät kauaksi optiselta akselilta, kuten kuvassa 15. Linssi on tällöin ellipsin muotoinen, jolloin valo taittuu erillä tavalla linssin ylä- ja alaosaan kuin linssin vasemmalla tai oikealla reunalla eikä ole siis symmetrinen. Tämä taittaa valon pois optiselta akselilta ja aiheuttaa sumeutta joko vertikaalisessa tai horisontaalisessa suunnassa tai kummassakin yhtä aikaa, mutta eri määrän.



Kuva 15. Hajataitto.

3.1.4 Petzval-kentän kaarevuus

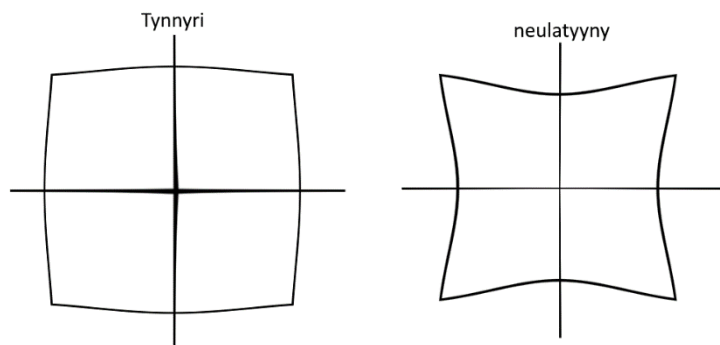
Olettaen että edelle mainittuja aberraatioita ei ole systeemissä ja kuva on täydellisesti kohdistettu, kuva voi piirtyä kaarevalle tasolle (kuva 16) eli Petzval-tasolle. Jos kuva tarkennetaan optiselle akselille, kuva on sitä epätarkempi, mitä suuremmassa kulmassa akseliin verrattuna valonsäteet tulevat. Pienen polttovälin linssit kärsivät enemmän tästä ongelmasta, koska näkökenttä on suuri.



Kuva 16. Kuva piirtyy pallomaiselle tasolle eli Petzval-tasolle. Optisen akselin suuntaiset säteet (sinisellä) tarkentuvat tasolle, mutta sivulta tulevat (punaisella) tarkentuvat pallon pinnalle.

3.1.5 Vääristymät

Geometriset vääristymät eivät aiheuta sumeutta kuvissa kuten muut aberraatiot. Vääristymissä polttopiste ja suurennos linssin eri kohdissa vaihtelee, jolloin kuvan muoto vääristyy, kuten kuvassa 17. Negatiivista vääristymää kutsutaan tynnyrivääristymäksi ja positiivista neulatyynyvääristymäksi.



Kuva 17. Tynnyri- ja neulatyynyvääristymä.

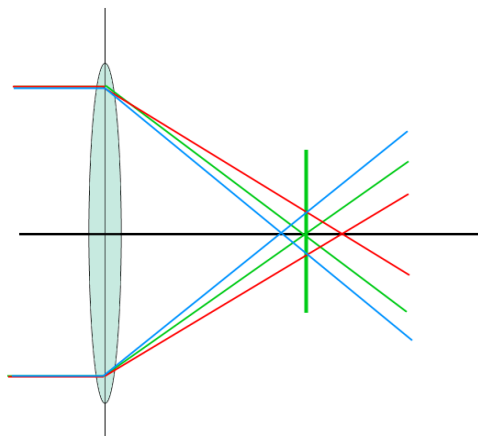
3.2 Kromaattinen aberraatio

Kromaattinen aberraatio johtuu valon eri aallonpituuksien taittumisesta kameran linssissä [17, 18, 19]. Tämä tapahtuu erityisesti paksuissa linseissä. Koska punainen, vihreä ja sininen aallonpituus taittuvat eri määrän, niiden polttopiste on eri. Jotta saataisiin täysin tarkka kuva, täytyisi linssi tarkentaa joka värikanavalle erikseen. Virheet tapahtuvat kuvan reuna-alueilla. Mitä kauemmas kuvan keskipisteestä päädytään, sen isompi virhe on.

Kromaattinen aberraatio jaetaan kahteen eri ilmiöön: pitkittäinen ja sivuttainen aberraatio. Valokuvauksessa esiintyy kumpaakin ilmiötä ja algoritmeissa niiden poistamisessa keskitytään jommankumman ilmiön korjaamiseen tai ilmiöt ajatellaan yhtenä. Algoritmeista kerrotaan enemmän myöhemmässä osassa.

3.2.1 *pitkittäinen aberraatio*

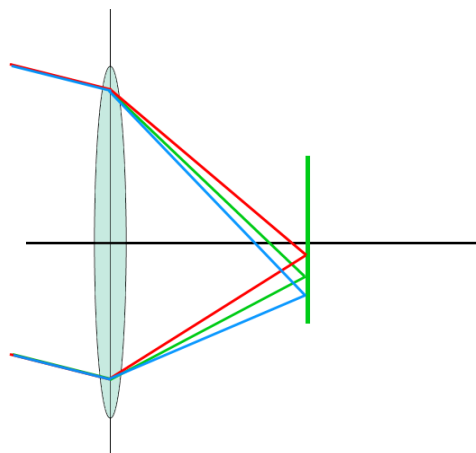
Pitkittäinen aberraatio voi tapahtua valolle, joka osuu linssiin samansuuntaisesti kuin linssin optinen akseli (kuva 18). Valon eri aallonpituudet taittuvat eri määrän ja niiden polttopiste on eri. Vain yhdestä värikanavasta saadaan tarkka ja loput kanavat ovat epätarkkoja. Pitkittäinen aberraatio aiheuttaa siten sumeutta kuvissa ja sitä on vaikea korjata jälkikäteen.



Kuva 18. Pitkittäinen aberraatio.

3.2.2 *Sivuttainen aberraatio*

Jälleen koska eri valon aallonpituudet taittuvat eri verran, eri värikanavat eivät aina ole yhtä tarkkoja. Sivuttaisessa aberraatiosta värikanavien polttopisteet ovat samalla viivalla horisontaalisesti, mutta eivät vertikaalisesti. Kuvasta 19 nähdään, kuinka valkoinen valo hajoaa ja värit kohdistuvat eri paikkoihin. Puhutaan niin sanotusta 'kromaattisesta erosta suurentamisessa' (eng. chromatic difference in magnification). Jos kamera kalibroidaan vihreälle kanavalle, sininen ja punainen ovat eri kokoisia vihreästä kuvasta (punainen kuva on pienempi ja sininen isompi).

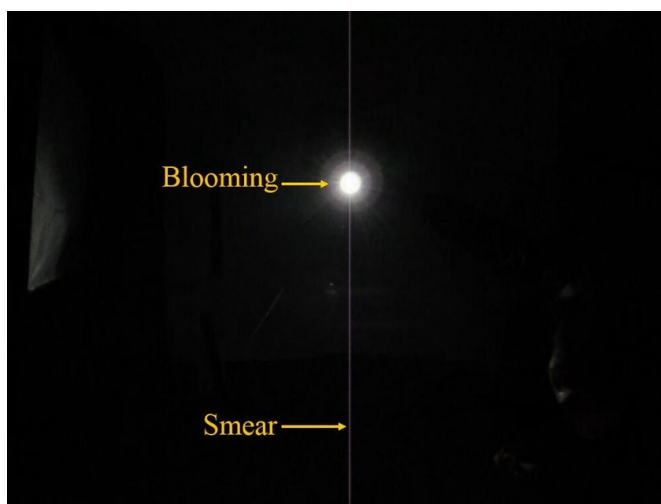


Kuva 19. Sivuttainen aberraatio.

Sivuttainen aberraatio aiheuttaa värien leviämistä ja siten myös violetteja reunoja. Toisin kuin pitkittäinen aberraatio, sivuttainen aberraatio on determinismistä ja se voidaan poistaa jälkikäteen ohjelmallisesti.

3.3 Pikseleiden ylivuoto

Linssi ei ole ainoa osa kamerasta, joka aiheuttaa artefakteja kuvissa. Kirkas valo voi aiheuttaa ylivuotoa viereisiin pikseleihin sensorissa. Ylivuoto näkyy joko ”blooming” tai ”smearing” -efektinä (kuva 20). Blooming tapahtuu, kun valo vuotaa sensorissa ympärillä oleviin pikseleihin ja smearing näkyy vertikaalisena viivana kuvissa. Monissa sensoreissa tämä on otettu huomioon ja sensoreihin on kehitetty sitä korjaavia osia sensoriin, mutta halvemmissä sensoreissa näitä virheitä voi vielä esiintyä.



Kuva 20. Sensorin ylivuotoa CCD -sensorissa.⁵

⁵ Lähde: Shen, Liang, et al. “Stroboscope Based Synchronization of Full Frame CCD Sensors.” Retrieved 18:21, May 12 2019 from <https://www.mdpi.com/1424-8220/17/4/799/htm> Available via license: CC BY 4.0

3.4 Interpolaatiovirheet

Suoraan kameran sensorilta saadut raakakuvat täytyy muuttaa RGB kuviksi ennen kuin ne voidaan näyttää tietokoneiden näytöillä. Raakakuva sisältää yhden pikselin kohdalla vain yhden värikanavan arvon, joten loput värit pitää interpoloida ympäristöstä. Tämä interpolaatio aiheuttaa virheitä varsinkin reuna-alueilla, joka näkyy joko vääränä värinä tai vetoketjuvirheenä [7, 8, 9, 19].



Kuva 21. Väärä väri (vasen) näkyy vihreänä värinä keltaisessa pyörässä ja vetoketjuvirhe (oikea) sahalaitaisena intensiteetin muutoksena.

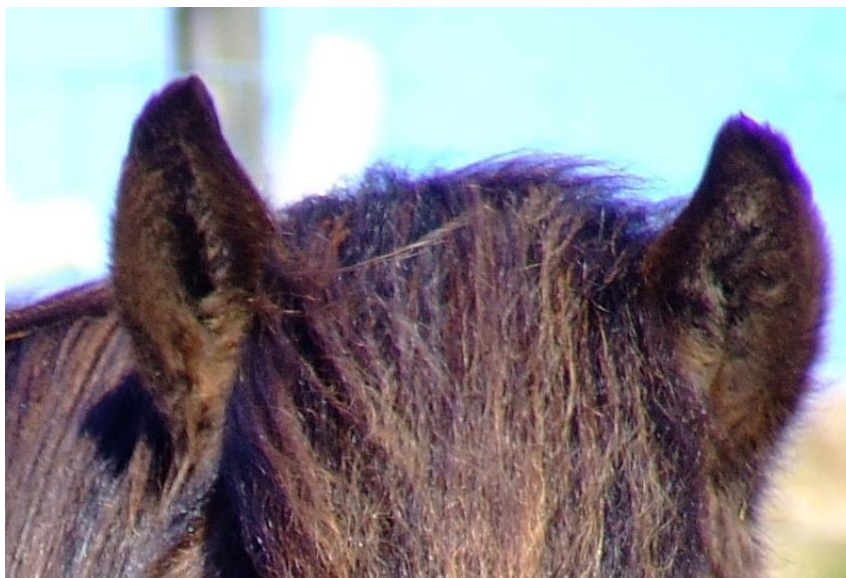
Esimerkiksi bilineaarinen interpolaatio aiheuttaa näitä virheitä, koska se ei ota huomioon reunoja, joissa värit muuttuvat terävästi. Bilineaarisessa interpolaatiossa puuttuvat arvot lasketaan 3×3 ympäristöstä ja kaikille paitsi reunapikseleillä löytyy 2 tai 4 pikseliä, joista R, G ja B voidaan laskea ottamalla näistä pikseleistä keskiarvo. Reuna-alue, jonka pitäisi olla terävä reuna, tasoitetaan interpolaatiolla ja sumeuttaa reunan. Jos esimerkiksi interpoloidaan reuna, jonka toinen puoli on tummansininen (0, 0, 128) ja toinen keltainen (255, 255, 128), saadaan näiden kahden pikselin väliin väriksi (128, 128, 128), joka on keskiarvo väreistä ja näkyy värivirheenä reuna-alueilla (kuva 21).

3.4 Violetti reunavirhe

Violetit reuna-alueet (eng. purple fringing) on sinisen ja punaisen värikanavan virheitä kuvassa. Virheen huomaa erityisesti suurissa kontrasteissa eli kohdissa, joissa vaaleat alueet muuttuvat tummiksi. Kuvasta 22 nähdään violettiä värivirhettä erityisesti alueissa, jotka ovat taivasta tai muuta vaaleaa taustaa vasten, mutta myös hieman muuallakin.

Violetti reunavirhe on monien virheiden yhdistelmä erityisesti halvoissa kameroissa. Suurin syy värivirheille löytyy linssistä ja kromaattisesta aberratiosta. Valon eri aallonpituuksien erilainen taittuminen aiheuttaa virhettä eniten kuvan reunoilla, mutta myös kuvan keskellä voi tapahtua virhettä, jos kuvan keskellä on paljon saturoitunutta aluetta eli valoisat alueet ovat ylivuotaneet ympäröiviin pikseleihin.

Värivirhe on violettia, koska virhe tapahtuu sinisissä ja punaisissa pikseleissä. Vihreää värivirhettä ei kuvissa yleensä ole tai se on hyvin vähäistä verrattuna punaiseen ja siniseen. Miksi vihreää värivirhettä ei tapahdu, johtuu monesta syystä. Kameran sensoreissa on enemmän vihreitä pikseleitä kuin punaisia tai sinisiä, joten vihreälle löytyy enemmän näytteitä, jonka ansiosta virheet suodattuvat pois. Kameran linssi voidaan myös kalibroida niin, että kuva on tarkka vihreälle kanavalle, mutta ei punaiselle ja siniselle. Lisäksi edellä mainitut kuvan artefaktit kaikki vaikuttavat lopulliseen värivirheeseen.



Kuva 22. Violettia reunavirhettä. ⁶

3.5 Kuvanlaadun parantavia ja korjaavia algoritmeja

Kuvanlaatua voidaan parantaa myös muillakin menetelmillä kuin käyttämällä parhaita mahdollisia linsejä tai sensoreita. Esimerkiksi mobiililaitteissa rajoituksina on koko ja hinta, jolloin saaduissa kuvissa tulee olemaan virheitä. Objektiivin ei voi olla samankokoinen kuin järjestelmäkameroissa. Näitä virheitä voidaan vähentää algoritmeilla jälkikäteen käsittelemällä. Uusissa älypuhelimissa ja jopa järjestelmäkameroissa on tehokas prosessori, jolla kuvaa voidaan laskennallisesti muokata ja korjata. Algoritmeilla parannetaan kuvan kontrastia, kohinaa ja edellä mainittuja artefakteja. Tässä kappaleessa keskitytään värivirheisiin ja kuinka ne voidaan tunnistaa ja poistaa kuvista.

Ensin käydään läpi kuitenkin gradientti. Monissa menetelmissä halutaan korjata violetti vain reuna-alueilta, joten tarvitaan keino löytää nämä reunat. Gradientissa lasketaan pikseleiden arvojen muutos ja muutoksen suunta pikselin ympäröivällä alueella. Koska kuva on kaksiulotteinen, muutos lasketaan sekä horisontaalisessa että vertikaalisessa suunnassa. Suuruus saadaan yhdistämällä nämä kaksi arvoa toisiinsa. Muutoksen laskemiseen kuvankäsittelyssä käytetään 3x3 ikkunaoperaattoreita, joilla

⁶ Lähde: File:Purple fringing.jpg. (2018, April 15). *Wikimedia Commons, the free media repository*. Retrieved 17:25, April 20, 2019 from https://commons.wikimedia.org/w/index.php?title=File:Purple_fringing.jpg&oldid=297258561.

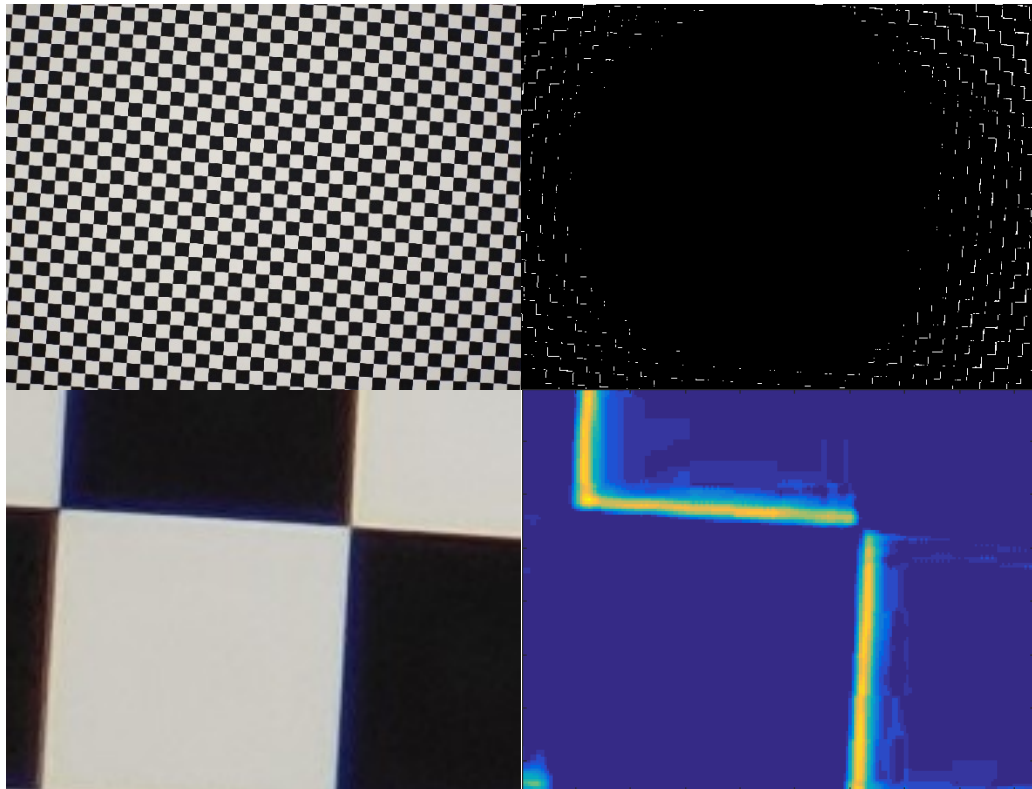
derivaattaa approksimoidaan. Kaksi yleistä operaattoria ovat Prewitt- ja Sobel-operaattori. Prewitt-operaattori ottaa viereiset pikselit ja summaa ne yhteen seuraavien painojen mukaan:

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +1 & 0 & -1 \\ +1 & 0 & -1 \end{bmatrix}, \quad G_y = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix},$$

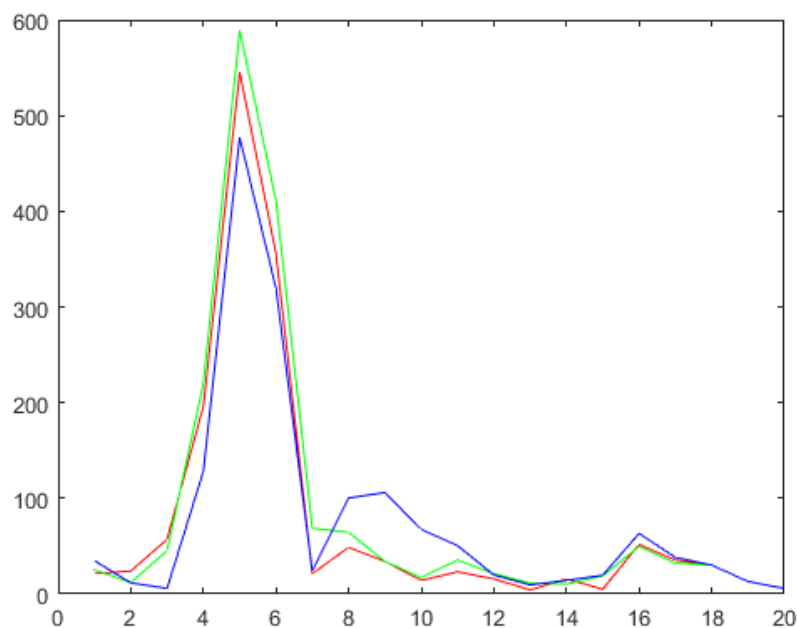
jossa G_x on horisontaalinen gradientti ja G_y vertikaalinen. Sobel-operaattori on lähes samanlainen, mutta keskimmäisiä pikseleitä painotetaan vähän enemmän.

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}, \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Kuvassa 24 on kuvasta 23 otettu horisontaalinen gradientti jokaiselle värikanavalle alueesta, jossa violettiä värivirhettä esiintyy. Kaikkien kanavien suurin muutos tapahtuu valkoisen muuttuessa mustaksi. Sininen ja punainen kanava pienenevät vihreää kanavaa vähemmän ja sininen muuttuu taas vähän isoimman muutoksen jälkeen. Reunan kohdalla on siis vielä sinistä ja vähän punaista, joka muuttuu mustaksi, kun reunasta on kuljettu pieni matka.



Kuva 23. Värivirhettä shakkilautakuvassa. Ylhäällä vasemmalla alkuperäinen kuva, ylhäällä oikealla binäärikuva, jossa värivirheet merkattu valkoisella, alhaalla vasemmalla suurennettu kuva värivirheestä (sininen hohto mustalla alueella) ja alhaalla oikealla sinisen värikanavan erotus mustavalkoisesta kuvasta.



Kuva 24. Leikkaus kuvan värien gradientti reuna-alueella.

3.5.1 Kang

Kangin algoritmi [20] korjaa violetin reunavirheen kahdessa osassa: ensin kuvasta etsitään alueet, jotka ovat kirkkaita ja sitten vertaillaan värikanavien eroja ja päätetään, onko alue värivirhettä. Koska violetti värivirhe tapahtuu useimmiten kirkkaiden alueiden ympärillä, etsitään kuvasta nämä kirkkaat alueet kuvan intensiteetin avulla. Intensiteetti saadaan helposti harmaakuvasta, joka lasketaan ottamalla keskiarvo R, G ja B arvoista. Kaikki pikselit, joiden intensiteetti on suurempi kuin annettu kynnyksarvo, lasketaan lähes-saturoituneeksi pikseliksi. Kang käyttää arvoa 230 kynnyksenä esimerkissään, mutta arvo voi olla mikä tahansa suuri arvo.

Toisessa vaiheessa kuvasta etsitään mahdolliset värivirhealueet vertailemalla värikanavien eroja. Pikseli lasketaan ehdokkaaksi, jos se toteuttaa kaksi ehtoa:

$$(B - R) > 25 \text{ ja } (B - G) > 25$$

Jos pikseli on ehdokas ja se on lähes-saturoitunut, se lasketaan violetiksi värivirheeksi ja sen violettisuutta vähennetään asettamalla sinisen ja punainen samaksi kuin vihreä eli tehdään pikselistä harmaa vihreän mukaan. Tämä on hyvin yksinkertainen menetelmä, joka poistaa pahimmat virheet, mutta voi poistaa myös oikeaa väriä, koska se ottaa huomioon vain pikselin värin eikä esimerkiksi pikselin ympäristöä.

3.5.2 Soon-Wook Chung & al

Soon-Wook Chung & al [21] ottaa huomioon myös pikselin ympäristön laskemalla koko kuvasta mahdolliset virhealueet gradientilla. Ideana on etsiä pikseli, jossa

tapahtuu paljon muutoksia, jonka jälkeen pikselistä ympäristöstä etsitään värivirhealue. Ensin kuvasta jokaisesta värikanavasta tehdään gradienttikuva yleisesti käytetyllä Sobelin operaattorilla.

$$E_{C_x}(i, j) = C(i-1, j-1) + 2C(i, j-1) + C(i+1, j-1) - C(i-1, j+1) - 2C(i, j+1) - C(i+1, j+1)$$

$$E_{C_y}(i, j) = C(i-1, j-1) + 2C(i-1, j) + C(i-1, j+1) - C(i+1, j-1) - 2C(i+1, j) - C(i+1, j+1),$$

jossa C on joko R , G tai B . Näistä horisontaalisesta ja vertikaalisesta gradientista lasketaan suuruus ja suunta jokaiselle värikanavalle $C \in \{R, G, B\}$:

$$E_C(i, j) = |E_{C_x}(i, j)| + |E_{C_y}(i, j)|,$$

$$\theta_C(i, j) = \arctan\left(\frac{E_{C_x}(i, j)}{E_{C_y}(i, j)}\right)$$

Kuvasta etsitään tarpeeksi suuri gradienttiarvo, jonka jälkeen kuljetaan joko pysty- tai vaakasuuntaan, kunnes saavutetaan tasainen, homogeeninen alue tai toinen reuna-alue kummassakin suunnassa. Esimerkiksi, jos kuljetaan kuvaa vaakasuunnassa, suuren gradientin kohdalla aletaan kulkemaan sekä vasemmalle että oikealle, kunnes löydetään pikselit, joissa seuraavan pikselin gradientti on kynnyksarvoa T_1 pienempi tai suurin kolmen värikanavan gradientteista on kynnyksarvoa T_2 pienempi:

$$H_x(j) = \max(|E_R(j)|, |E_G(j)|, |E_B(j)|)$$

$$l_k = j, \quad \text{jossa } j: \text{tä vähennetään kunnes } H_x(j) < T_2 \text{ tai } E_G(j) < T_1 \\ \leq E_G(j-1)$$

$$r_k = j, \quad \text{jossa } j: \text{tä kasvatetaan kunnes } H_x(j) < T_2 \text{ tai } E_G(j) < T_1 \\ \leq E_G(j+1)$$

jossa l_k on vasen reuna-alueen pikseli, r_k on oikean reuna-alueen pikseli ja $E_G(j)$ on vihreä gradientti. Näiden kahden pisteen välinen alue lasketaan potentiaalisesti värivirheeksi ja aletaan etsimään uusia samanlaisia alueita samalla tavalla. Potentiaalinen värivirhe lasketaan oikeaksi värivirheeksi, jos punaisen ja sinisen ero on liian pientä tai suurta verrattuna vasempaan tai oikeaan reunapisteseen.

$$D_r = R - G$$

$$D_b = B - G$$

$$\rho_s(j|k) = \begin{cases} 1, & \text{jos } D_s(j) < \min(D_s(l_k), D_s(r_k)) \text{ tai } D_s(j) > \max(D_s(l_k), D_s(r_k)) \\ 0, & \text{muulloin} \end{cases}$$

jossa $S \in \{R, B\}$ ja $j|k$ on l_k ja r_k välinen reuna-alue. Värivirheet poistetaan lopuksi tasoittamalla punaisen ja sinisen ero vihreästä samalle tasolle kuin reunapisteissä l_k tai r_k , jos $\rho_S(j|k)$ pikselin kohdalla on 1.

$$S_{out}(j) = G_{in}(j) + \min(D_S(r_k), D_S(l_k))$$

3.5.3 Kim ja Park

Kim ja Park [22] etsivät lähes-saturoituneet pikselit samalla tavalla kuin Kang, mutta ehdokkaat lasketaan hieman toisella tavalla. Sinistä ja punaista värikanavaa verrataan vihreään, ja jos ne ylittävät tai alittavat kynnyksarvon, lasketaan alue mahdolliseksi värivirheeksi:

$$CR'(x,y) = \begin{cases} 1, & \text{jos } I_r(x,y) - I_b(x,y) < T \text{ ja } I_b(x,y) - I_g(x,y) > T \\ 0, & \text{muuten} \end{cases},$$

jossa I_r , I_b ja I_g ovat värikanavien intensiteetit ja T on sama kynnyksarvo kuin Kangin algoritmissa. Lisäksi otetaan huomioon värikanavien gradientit, mutta kuvaa ei käydä läpi yhtä tarkasti kuin Soon-Wook Chung & al menetelmässä. Kuvan värit kvantisoidaan ensin neljään tasoon, jotta prosessi ei olisi liian aikaa vievä:

$$QI_c(x,y) = \begin{cases} 3, & \frac{7}{8}(I_{cmax} - I_{cmin}) < I_c(x,y) \leq I_{cmax} \\ 2, & \frac{3}{4}(I_{cmax} - I_{cmin}) < I_c(x,y) \leq \frac{7}{8}(I_{cmax} - I_{cmin}) \\ 1, & \frac{1}{2}(I_{cmax} - I_{cmin}) < I_c(x,y) \leq \frac{3}{4}(I_{cmax} - I_{cmin}) \\ 0, & (I_{cmin}) < I_c(x,y) \leq \frac{1}{2}(I_{cmax} - I_{cmin}) \end{cases}$$

Tämän jälkeen tästä kvantisoiduista kuvasta lasketaan binäärikuva, jossa kaikki pikselit, jossa jokin kvantisoidun värikanavan gradientti on suurempi kuin nolla, asetetaan ykköseksi. Gradientti lasketaan joka värikanavalle ja muutetaan binääriksi kaavoilla:

$$| \nabla QI_c(x,y) | = \sqrt{\left(\frac{dQI_c(x,y)}{dx} \right)^2 + \left(\frac{dQI_c(x,y)}{dy} \right)^2}$$

$$BGM(x,y) = \begin{cases} 1, & \text{jos } | \nabla QI_R(x,y) | \neq 0 \text{ tai} \\ & | \nabla QI_G(x,y) | \neq 0 \text{ tai} \\ & | \nabla QI_B(x,y) | \neq 0 \\ 0, & \text{muulloin} \end{cases}$$

Lopulta alueet, jotka ovat saturoituneita ja joiden gradientti on suurempi kuin nolla, korjataan muuttamalla alue harmaaksi eli kaikki värikanavat asetetaan sen kanavien keskiarvoksi:

$$I_C(x, y) = \frac{I_R(x, y) + I_G(x, y) + I_B(x, y)}{3}$$

Tällä menetelmällä korjataan vain myös alueet, joissa punainen on sinistä suurempi, toisin kuin Kangin menetelmässä, jossa nämä alueet jätetään korjaamatta. Gradientilla saadaan taas korjattua vain reuna-alueet, jolloin tasaisen violetit alueet jätetään rauhaan.

3.5.4 Kannan ja Malik

Kaikki ylemmät algoritmit toimivat vain RGB väriavaruudessa. Jos korjattava kuva olisi YCrCb formaattia täytyisi kuva muuttaa RGB:ksi ja takasin. Kannan ja Malik [23] toteuttavat reunavirheiden poiston YCrCb-väriavaruudessa. Periaate on samanlainen kuin aikaisemmissa menetelmissä: kuvasta etsitään saturoituneet alueet ja gradientin avulla lasketaan pikselit, joissa kontrasti on suurta. Gradientti ja saturaatio lasketaan vain Y eli luminanssikanavasta. Gradienttikuvasta lasketaan gradienttien keskihajonta, jonka avulla selvitetään kynnsarvo T_{grad} , jolla alue lasketaan saturoituneeksi alueeksi:

$$T_{grad} = \mu_{grad} + \alpha\sigma$$

$$M_Y(x, y) > T_{grad}$$

missä μ_{grad} on gradienttien keskiarvo, σ on gradienttien keskihajonta, α on valittu painokerroin välillä $[0, 3]$ ja $M_Y(x, y)$ on gradientin suuruus pisteettä (x, y) .

3.5.5 Algoritmien yhteenveto

Listatut algoritmit poistavat värivirhettä tunnistamalla kuvasta alueita, joissa tapahtuu paljon muutoksia tai ovat kirkkaita ja näin ollen mahdollisesti esiintyy värivirheitä. Nämä alueet tunnistetaan joko gradientilla, kynnsarvoilla tai niiden yhdistelmällä. Värivirheet löydetään alueilta vertaamalla sinistä ja punaista väriä vihreään. Kun värivirhe on löydetty, muutetaan väri vähentämällä sinistä ja punaista. Vaikka algoritmit eroavat toisistaan sillä, miten ne päätyvät näihin päätöksiin, ovat ne samanlaisia vertaamalla vihreää värikanavaa jollakin tavalla ja etsimällä alueita, joissa on muutoksia. Ongelmana algoritmeissa on, että ne poistavat vain pahimmat värivirheet (varsinkin hyvin violetit virheet) ja eivät poista enemmän sinistä violettia. Kynnsarvot Kangin sekä Chungin algoritmeissa aiheuttavat katkeavaa korjausta, jolloin väri korjataan yhdestä kohdasta, mutta ei sen vierestä.

4 TOTEUTUS

Vaikka älypuhelimet ovat kehittyneet jo hyvin nopeiksi, on niiden nopeus vielä kaukana pöytätietokoneista. Sen takia puhelimiin suunnitellut algoritmit pitävät olla yksinkertaisia, jotta niistä saadaan tarpeeksi nopeita. Tämä toteutus keskittyy nopeuteen, jolloin tarkkuus voi kärsiä paikoitellen. Samalla kuitenkin algoritmi poistaa enemmän virheitä alueilla, jotka eivät ole yhtä kirkkaita kuin Kangin [20] algoritmossa. Kangin algoritmi poistaa lähes saturoituneita pikseleitä, joiden arvo on > 230 , mutta tässä työssä käytetyissä kuvissa voi olla sinertävää virhettä pikseleissä, joiden kirkkaus on vain 150. Suurin osa virheistä jää siis poistamatta Kangin algoritmilla, jos violetti reunavirhe on pahasti sotkenut alueita, mutta toisaalta virheellisiä korjauksia tapahtuu vähemmän.

Toteutus esitetään tässä kohdassa pseudokoodina kohta kerrallaan. Koko pseudokoodi löytyy liitteestä 1. Toteutus jaetaan kahteen osaan: värin korjaukseen ja maskin luontiin. Maski kertoo, mistä väriä pitää korjata ja kuinka paljon, joka lasketaan gradientin avulla. Värien korjaus taas tapahtuu vähentämällä sinistä ja punaista värikanavaa samalla seuraten kolmea sääntöä:

1. $B \geq G$ eli sinistä ei vähennetä vähemmäksi kuin vihreää
2. $R \geq G$ eli punaista ei vähennetä pienemmäksi kuin vihreää
3. $R/B > \text{vakio}$ eli punaisen ja sinisen suhde pysyy määrätyn vakion yläpuolella

Korjauksessa siis vähennetään sinistä ja punaista samalle tasolle kuin vihreä (harmaasävy), mutta säilyttäen sinisen ja punaisen välisen suhteen jonkin rajan yläpuolella. Ensin lasketaan R :n ja B :n ero vihreästä.

$$\begin{aligned} dR &= \max(0, R - G) \\ dB &= \max(0, B - G) \end{aligned}$$

Harmaasävy saataisiin vähentämällä ero alkuperäisestä arvosta eli $R - dR$ ja $B - dB$, mutta suhde halutaan säilyttää. Tätä varten sinistä värikanavaa sumennetaan:

$$B_2 = \max(0, (B - thr)) * \left(\frac{255}{255 - thr}\right)$$

$$B_2 = \text{boxblur}(B_2, [5, 5])$$

jossa B on sininen värikanava, thr on kynnyisarvo, jolla kuvan tummia alueita kirkastetaan, B_2 on sumennettu kuva ja boxblur on keskiarvoistava suodatin, joka ottaa pikselin 5x5 ympäristöstä keskiarvon. B_2 avulla lasketaan minimiarvo, jolla värisuhde vielä säilyy.

$$\begin{aligned} mB &= \min(B_2, dB) \\ \Delta R &= \min(dR, mB * \max RB) \end{aligned}$$

$$\Delta B = \begin{cases} \min(mB, dR / \min RB), & \text{kun } \min RB > 0 \\ mB, & \text{kun } \min RB = 0 \end{cases}$$

jossa maxRB on maksimisuhde punaisen ja sinisen välillä ja minRB on pienin suhde. Toteutuksessa minRB on 0 ja maxRB on 1/3, jolloin punaisen vähennystä on rajoitettu. Lopulta uusi R' ja B' saadaan vähentämällä ΔR ja ΔB alkuperäisistä arvoistaan:

$$\begin{aligned} R' &= R - \Delta R \\ B' &= B - \Delta B \end{aligned}$$

Jos koko kuva korjataan tällä tavalla, muutetaan kaikki sinertävät tai violetit alueet vihertäviksi, jolloin esimerkiksi taivas tai liikennemerkit muutetaan virheellisesti (kuva 25 oikea yläkulma). Jotta vältetään virheet, lasketaan vielä maski käyttämällä laskettuja muutoksia.

Muuttujista ΔR ja ΔB lasketaan gradientti maskia varten. Gradientti lasketaan käyttämällä Sobelin operaattoria, kuten kohdassa 3.5.2. Vain gradientin suuruus merkitsee, joten horisontaalinen gradientti E_x ja vertikaalinen gradientti E_y yhdistetään kaavalla $E_C = \sqrt{(E_x^2 + E_y^2)}$, jossa C on joko R tai B. Näin saadaan gradienttikuvat E_R ja E_B , jotka normalisoidaan välille [0 1] ja yhdistetään yhdeksi:

$$normE_R = \max(0, \min\left(1, \frac{E_R - minRgradient}{maxRgradient}\right))$$

$$normE_B = \max(0, \min\left(1, \frac{E_B - minBgradient}{maxBgradient}\right))$$

$$E = (normE_R + 2 * normE_B) / 3$$

jossa maxRgradient ja maxBgradient ovat kynnyisarvoja, joita suuremmat gradientin arvot lasketaan ykköseksi ja minRgradientia ja minBgradientia pienemmät arvot lasketaan nolaksi. Sinisestä gradientista painotetaan enemmän kuin punaisesta, koska värivirhettä on enemmän sinisessä värikanavassa. Kuvassa 25 löytyy gradientin E kuva vasemmasta alakulmasta.

Gradientilla saadaan estettyä isojen, tasaisten alueiden, kuten taivaan, virheellinen korjaus, mutta taivaan reuna-alueet (pilvien reunat, talojen katot) jäävät vielä vihreiksi. Hyvin tasaiset alueet löydetään sumentamalla sininen värikanava ja vähentämällä siitä alkuperäinen sininen. Alueet, joissa erot ovat pieniä, lasketaan tasaisiksi alueiksi ja korjausta vähennetään näiltä alueilta.

$$eroB = boxblur(B, [5, 5]) - B$$

$$tasainenB = \frac{abs(eroB) - minB}{maxB}$$

Erotuksen eroB itseisarvolle tehdään samanlainen kynnystys kuin gradientille ja saadaan tasainenB, jossa tasaiset alueet ovat lähellä nollaa ja suurten muutoksien alueet ovat lähellä yhtä. Lopullinen tulos, jonka näkee kuvassa 25 oikeassa alakulmassa, saadaan yhdistämällä gradientti, tasainenB ja lasketut ΔR ja ΔB .

$$\begin{aligned} korjattuR &= R - \Delta R * tasainenB * E \\ korjattuB &= B - \Delta B * tasainenB * E \end{aligned}$$

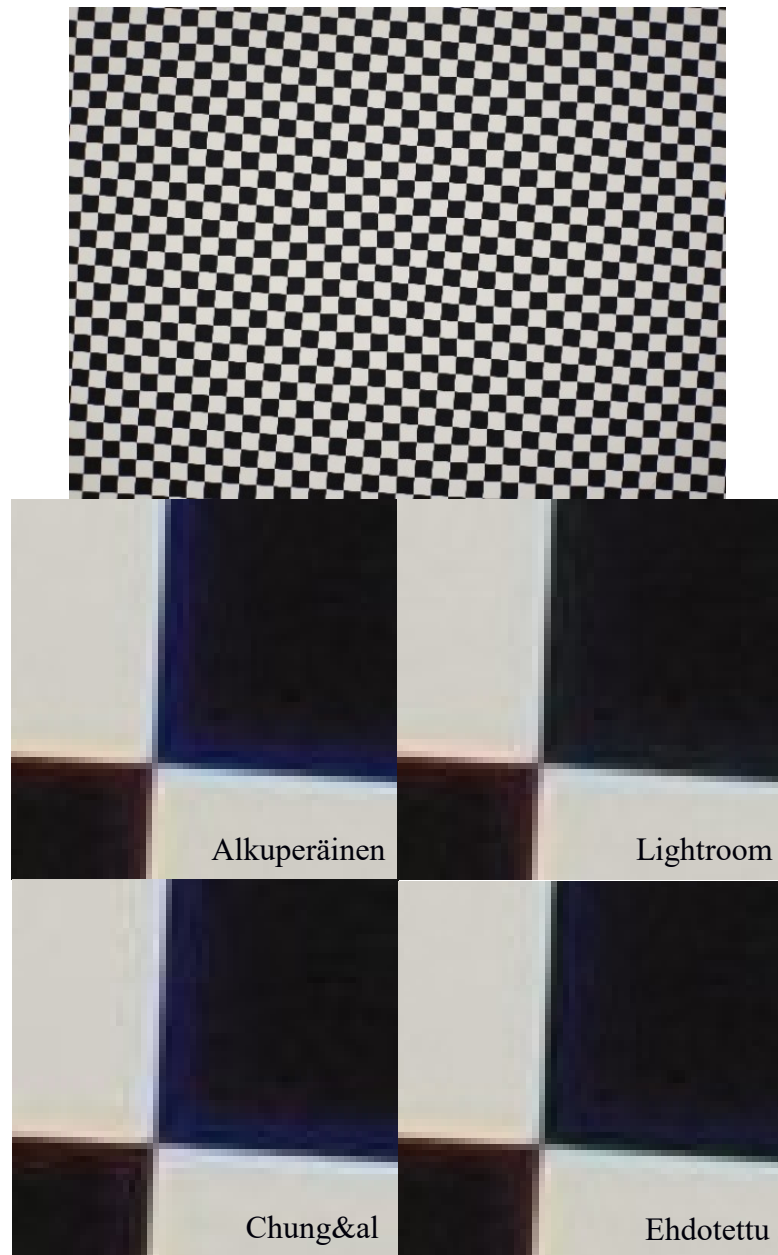
Vain punainen ja sininen väri korjataan ja vihreä jätetään alkuperäiseksi.



Kuva 25. Oikealla ylhäällä alkuperäinen kuva. Vasemmalla ylhäällä värikorjaus kuvalle ilman maskia. Alhaalla oikealla gradienttikuva ja oikealla alhaalla lopullinen kuva. Violetti väri on poistettu lehdistä (esim. punaisen laatikon sisällä).

5 TULOKSET

Violettia reunavirhettä ei aina voida poistaa kokonaan, koska se johtuu monesta tekijästä ja riippuu myös laitteesta, joten siitä on vaikea irrottaa perustotuutta, johon tulosta voitaisiin verrata. Niinpä voidaan tuloksia verrata esimerkiksi käyttämällä mustavalkoista shakkilautaakuviota, jossa tiedetään, että väriä ei pitäisi esiintyä. Normaaleissa kuvissa on kuitenkin paljon väriä, jolloin matemaattisesti on vaikea kertoa, kuinka paljon kuvaa on parannettu. Siksi tuloskuvia arvioidaan kahdella tavalla: vertaamalla kuinka paljon kuva eroaa mustavalkoisesta shakkilaudasta (mitä enemmän eroaa, sitä enemmän virheitä on tullut) sekä kyselyllä, jossa kysytään, kumpi kuva on vastaajasta parempi.



Kuva 26. Shakkilautakuva ja tuloskuvat. Suurennetuista tuloskuvista näkee eri algoritmien vaikutukset.

5.1 Tulosten ero mustavalkokuvasta

Shakkilautakuvan (kuva 26) pitäisi olla lähes täysin mustavalkoinen, joten kaiken värin voi olettaa olevan virhettä. Koska kuvassa ei ole hyvin kirkkaita kohteita, kuten aurinkoa, värivirheen voi myös olettaa johtuvan täysin kromaattisesta aberratiosta. Mustavalkoinen kuva saadaan ottamalla värikanavien keskiarvo, jota sitten verrataan tulokkuviin laskemalla erojen neliöiden summa ja jakamalla pikselien määrällä:

$$\text{värivirhe} = \frac{\text{sum}((BW - I_c)^2)}{\text{leveys} * \text{korkeus}}$$

jossa BW on mustavalkoinen kuva ja I_c on yhden RGB värikanavan kaikki pikselit. Mitä lähempänä luku on nollaa, sen paremmin algoritmi poistaa värivirheitä. Mustavalkoista kuvaa BW verrataan alkuperäiseen kuvaa, omaan menetelmään, Adobe Lightroomin kuvaan sekä Soon-Wook Chung & al [21] menetelmään. Taulukosta 1 näkee, että Adobe Lightroom pystyy poistamaan värivirhettä parhaiten, kun taas ehdotettu menetelmä poistaa toiseksi parhaiten. Chungin menetelmä ei poista tästä kuvasta juurikaan sinistä, mutta poistaa kyselyssä käytetyistä kuvista paremmin erottuvia värivirheitä.

Taulukko 1. Mustavalkoisen kuvan ja värikanavien keskimääräinen ero per pikseli

	alkuperäinen	ehdotettu	Lightroom	Soon-Wook Chung
R	9,0736	9,080	10,7633	8.0297
G	0,9754	0,9754	2,1523	0.9754
B	47,2591	36,1932	27,6961	46.8367

5.2 Tulosten arviointi kyselyllä

Kyselyssä verrataan tämän työn toteutusta kahteen eri toteutukseen sekä käsittelemättömään kuvaan. Toisina toteutuksina käytetään Soon-Wook Chung & al [21] toteutusta sekä kaupallista Adobe Lightroomin automaattista kromaattisen aberration poistoa. Kyselyn vastaajia pyydetään sitten vertaamaan kahta kuvaa, joista valitaan parempi kahdesta. Näin saaduilla tuloksilla saadaan subjektiivinen käsitys, kuinka hyvä tai huono toteutus on verrattuna muihin algoritmeihin. Kyselyssä verrataan menetelmiä vertailemalla kahta menetelmää (tai alkuperäistä) keskenään. Kyselyn vastaaja näkee tietokoneen ruudulla yhden kuvan, jonka hän pystyy vaihtamaan toiseen, vertailtavaan kuvaan. Vastaaja päättää kolmesta vaihtoehdosta: kuva 1 on parempi, kuva 2 on parempi tai kuvat näyttävät samalta. Kuvia on 12 ja menetelmien yhdistelmiä on 6, joten kyselyn aikana käydään läpi 72 kuvaparia.

Kyselyyn vastasi 10 henkilöä, jolloin valintoja on yhteensä 720. Kuvia, joita vastaajille näytettiin, on havainnollistettu kuvassa 27. Tulokset ovat nähtävissä taulukosta 2. Vastaukset jakaantuivat tasaisesti, joten korjaukset joko onnistuivat tai eivät. Jos valintoja tarkastelee tarkemmin, kuvissa, joita ei valittu, näkee selkeitä väärin korjattuja alueita. Vaikka kuvista poistetaan värivirhettä, väärin korjattuja alueita ei suvaittu juuri ollenkaan, kuten liikennemerkkien reunojen sotkemista. Osassa kuvissa eri menetelmä korjaa värin eri tavalla. Esimerkiksi Lightroom korjattu

väri on sinisempää kuin ehdotetussa menetelmässä, joka on selvästi vihreä. Tällöin valittiin sinisempi vaihtoehto, joka korjaa virheet hienovaraisemmin (erityisesti kuvissa, jossa näkyy puiden oksia sinistä taivasta vasten).

Taulukko 2. Kyselyn vastaajien valinnat kuvapareja vertaillessa.

kuva 1 – kuva 2	1 parempi	2 parempi	sama
alkuperäinen - ehdotettu	52	54	14
alkuperäinen - Chung	43	54	23
alkuperäinen – Lightroom	37	38	45
Lightroom – ehdotettu	57	43	20
Lightroom – Chung	67	49	4
ehdotettu - Chung	46	62	12



Kuva 27. Kyselyssä käytettyjä kuvia.

6 KESKUSTELU

Violettia reunavirhettä on vaikea korjata objektiivisesti, koska se johtuu monesta asiasta, kuten kromaattisesta aberraatiosta tai ylivalottuneista alueista. Tuloksia on vaikea arvioida, koska pohjatotuutta, johon tuloksia voisi verrata, on vaikea luoda. Siksi tuloksien laatua arviotiin kyselyllä, mutta värivirheet voivat olla välillä niin pieniä, että vastaajien oli vaikea nähdä korjausten vaikutus. Ehkä siksi tulokset olivat hyvin jakaantuneita eikä hyvää vertailua löytynyt. Kyselyn voisi tehdä hyvin tarkaksi, jossa näytetyt kuvat olisivat hyvin paljon suurennettuja. Silloin kuvasta ei tosin erota enää, mitä se esittää, jolloin tulokset voivat vääristyä.

Toinen ongelma violetin värivirheen korjauksessa on virheelliset korjaukset, jotka sotkevat kuvaa. Koska korjauksille ei ole täydellistä fysikaalista mallia, joudutaan kuvaa korjaamaan värin perusteella, mutta joskus väri voi olla oikeaa eikä virhettä, jolloin korjataan esimerkiksi liikennemerkkejä. Kyselyssä, jos liikennemerkkit olivat sotkeentuneet, valittiin mieluummin kuva, jota ei oltu korjattu. Jos taas korjauksen vahvuutta pienennetään, korjaukset ovat niin pieniä, ettei niitä enää huomata. Käytetyissä testikuvissa oli myös hyvin sinertävää värivirhettä sekä violettia värivirhettä. Osa sinisestä voi olla pelkkää kromaattista aberraatiota, kuten shakkilautakuviossa, mutta se ei selitä joidenkin kuvien taivaan ja puiden välistä reunaa. On siis vaikea tarkasti korjata väriä, koska väri voi vaihdella kuvien välillä.

Eri menetelmät myös korjasivat väriä muuttamalla värin joksikin toiseksi. Helpoin ratkaisu on muuttaa violetti suoraan harmaasävyksi, mutta miksi väriksi violetti tai sininen värivirhe pitäisi muuttaa. Kauempaa katsottuna kuva voi muuttua vihertäväksi, joka näyttää huonolta taivasta vasten. Vaikka kaikki värivirheet löydettäisiinkin, pitäisi päättää alueelle oikeanlainen väri, joka sopii kuvaan.

7 YHTEENVETO

Tässä työssä käytiin läpi, miten kuva muodostuu kamerassa, mitä mahdollisia virheitä voi tulla kuvan luonnissa, muutamia kuvan korjausalgoritmeja ja ehdotettu toteutus. Aluksi käytiin läpi kameran linsejä, sensori ja muita kameran osia, jonka jälkeen listattiin mahdollisia kuvan virheitä. Jo olemassa olevia algoritmeja esitettiin 4 sekä esitettiin oma toteutus. Tuloksissa verrattiin ehdotettua toteutusta, yhtä olemassa olevaa algoritmia sekä kaupallista tuotetta toisiinsa kyselyn ja mustavalkoisen kuvan avulla. Kyselyn tulokset olivat sekalaisia eikä selvästi parasta algoritmia löytynyt. Mustavalkoisen kuvan vertailussa oma toteutus vähentää kuvasta virhettä paremmin kuin Chung, Soon-Wookin [21] menetelmä, mutta jää toiseksi Adobe Lightroomin korjaukselle.

8 LÄHTEET

- [1] Adams, Ansel. "The Camera, The Ansel Adams Photography Series." Little, Brown and Company 1981 (1980): 1983.
- [2] David Balihar - What is a pinhole camera? <https://www.pinhole.cz/en/pinholecameras/whatis.html>
- [3] R. Szeliski: Computer Vision: Algorithms and Applications, Springer, 2011.
- [4] Hecht, Eugene. Optics. Reading, Mass.: Addison-Wesley,, 2002.
- [5] Ma, Yuke, and V. Borovytsky. "Design of a 16.5 megapixel camera lens for a mobile phone." Open Access library journal2.3 (2015).
- [6] Nakamura, Junichi, ed. Image sensors and signal processing for digital still cameras. CRC press, 2016.
- [7] Ramanath, Rajeev, et al. "Demosaicking methods for Bayer color arrays." Journal of Electronic imaging 11.3 (2002): 306-316.
- [8] Kaur, S., and Vijay Kumar Banga. "A survey of demosaicing: Issues and challenges." International Journal of Science, Engineering and Technologies 2.1 (2015): 2.
- [9] Losson, Olivier, Ludovic Macaire, and Yanqin Yang. "Comparison of color demosaicing methods." Advances in Imaging and Electron Physics. Vol. 162. Elsevier, 2010. 173-265.
- [10] Stanford university, The Camera Image Processing Pipeline (2018), [luettu 5.5.2019] http://graphics.stanford.edu/courses/cs348v-18-winter/lectures/02_camerapipeline.pdf
- [11] Stanford university, Kari Pulli, Camera Processing Pipeline, [luettu 5.5.2019] <https://web.stanford.edu/class/cs231m/lectures/lecture-11-camera-isp.pdf>
- [12] Optical Objective Systems: A Schematic Guide
- [13] Fischer, Robert Edward, et al. Optical system design. Vol. 1. No. 2. New York: McGraw Hill, 2000.
- [14] Dereniak, Eustace L., and Teresa D. Dereniak. "Geometrical and trigonometric optics." Geometrical and Trigonometric Optics, by Eustace L. Dereniak, Teresa D. Dereniak, Cambridge, UK: Cambridge University Press, 2008 (2008).
- [15] Smith, Warren J. Modern optical engineering. Tata McGraw-Hill Education, 1966.
- [16] Hecht, Eugene. Optics 5th edition. Pearson, 2017

- [17] Peatross, Justin, and Michael Ware. Physics of light and optics. Brigham Young University, Department of Physics, 2011.
- [18] Smith, Warren J. Modern optical engineering. Tata McGraw-Hill Education, 1966.
- [19] Hernández, Diana Carolina Morón, et al. "Quality Evaluation of Chromatic Interpolation Algorithms for Image Acquisition System." *Journal of Aerospace Technology and Management* 8.3 (2016): 339-351.
- [20] Kang, Sing Bing. "Automatic removal of purple fringing from images." U.S. Patent No. 7,577,292. 18 Aug. 2009.
- [21] Chung, Soon-Wook, Byoung-Kwang Kim, and Woo-Jin Song. "Detecting and eliminating chromatic aberration in digital images." *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2009.
- [22] Kim, Baek-Kyu, and Rae-Hong Park. "Automatic detection and correction of purple fringing using the gradient information and desaturation." *Signal Processing Conference, 2008 16th European*. IEEE, 2008.
- [23] Karthik, Kannan, and Parveen Malik. "Purple Fringing Aberration Detection Based on Content-Adaptable Thresholds." *Proceedings of First International Conference on Smart System, Innovations and Computing*. Springer, Singapore, 2018.

9 LIITTEET

Liite 1. Toteutuksen pseudokoodi

$$dR = \max(0, R - G)$$

$$dB = \max(0, B - G)$$

$$B_2 = \max(0, (B - thr)) * (\frac{255}{255 - thr})$$

$$B_2 = \text{boxblur}(B_2, [5, 5])$$

$$mB = \min(B_2, dB)$$

$$\Delta R = \min(dR, mB * \max RB)$$

$$\Delta B = \begin{cases} \min(mB, dR / \min RB), & \text{kun } \min RB > 0 \\ mB, & \text{kun } \min RB = 0 \end{cases}$$

$$\text{norm}E_R = \max(0, \min(1, \frac{E_R - \min R \text{gradient}}{\max R \text{gradient}}))$$

$$\text{norm}E_B = \max(0, \min(1, \frac{E_B - \min B \text{gradient}}{\max B \text{gradient}}))$$

$$E = (\text{norm}E_R + 2 * \text{norm}E_B) / 3$$

$$\text{ero}B = \text{boxblur}(B, [5, 5]) - B$$

$$\text{tasainen}B = \frac{\text{abs}(\text{ero}B) - \min B}{\max B}$$

$$\text{korjattu}R = R - \Delta R * \text{tasainen}B * E$$

$$\text{korjattu}B = B - \Delta B * \text{tasainen}B * E$$